smart_clustering_insight

October 16, 2020

1 S.Ma.R.T. CLUSTERING INSIGHT

[1]:
```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
```

```
<IPython.core.display.HTML object>
```

[2]:
```
import time
from datetime import timedelta
start_time = time.time()
```

[3]:
```
import joblib
import numpy  as np
import pandas as pd
import re
import scipy
```

[4]:
```
logs          = ["log_all_data.csv", "log_20200124.csv", "log_20200201.csv",␣
 ↪"log_20200202.csv", "log_20200203.csv", "log_20200204.csv"]
filename_orig = logs[0]
ruta          = "./dataset/" + filename_orig
fecha         = re.search('log_(.*)\.csv', filename_orig, re.IGNORECASE).
 ↪group(1)
filename_new  = "smart_predicted_style_{}.csv".format(fecha)
datos         = pd.read_csv(ruta)
df_ul         = datos.drop(['TagID_', 'DrillID_', 'TimeStamp_min',␣
 ↪'TimeStamp_max', 'TimeStampDT_min', 'TimeStampDT_max'], axis = 1)
```

[5]:
```
from sklearn.preprocessing import MinMaxScaler
min_max           = MinMaxScaler()
df_scaled         = pd.DataFrame(min_max.fit_transform(df_ul.astype("float64")))
df_scaled.columns = df_ul.columns
```

[6]:
```
model_clone     = joblib.load('smart_model.pkl')
model_clone.fit_predict(df_scaled)
datos_labels    = model_clone.labels_
datos_centroids = model_clone.cluster_centers_
```

[7]:
```
number_of_clusters      = 3
col_classifier          = list(df_scaled.columns).index("classifier_")
centroids_classifier    = [cc[col_classifier] for cc in datos_centroids]
index_min               = centroids_classifier.index(min(centroids_classifier))
index_max               = centroids_classifier.index(max(centroids_classifier))
lista_labels            = ["Others"] * number_of_clusters
lista_labels[index_min] = 'Breaststroke'
lista_labels[index_max] = 'Front Crawl/Freestyle'
```

[8]:
```
df_scaled['Prediction'] = datos_labels
df_scaled['Prediction'] = df_scaled['Prediction'].replace({0: lista_labels[0],
                                                           1: lista_labels[1],
                                                           2: lista_labels[2]})
df_scaled.groupby(['Prediction']).count().iloc[:, -1]
```

[8]:
```
Prediction
Breaststroke              402
Front Crawl/Freestyle    2884
Others                   1159
Name: classifier_, dtype: int64
```

[9]:
```
insight                       = pd.DataFrame(df_scaled.iloc[:, -1])
insight[['TagID', 'DrillID']] = datos[['TagID_', 'DrillID_']]
insight                       = insight[['TagID','DrillID','Prediction']]
insight.set_index(['TagID','DrillID'], inplace=True)
insight.to_csv(filename_new)
```

[10]:
```
elapsed_time_secs = time.time() - start_time
msg = "Execution took: %s secs (Wall clock time)" %␣
 ↪timedelta(seconds=round(elapsed_time_secs))
print(msg)
```

```
Execution took: 0:00:01 secs (Wall clock time)
```

[11]:
```
!jupyter nbconvert --to pdf smart_clustering_insight.ipynb
```

```
[NbConvertApp] Converting notebook smart_clustering_insight.ipynb to pdf
[NbConvertApp] Writing 32854 bytes to .\notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', '.\\notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', '.\\notebook']
[NbConvertApp] WARNING | b had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 29873 bytes to smart_clustering_insight.pdf
```

[ ]: